CLAIMS

What is claimed is:

1. A system that provides for remote password authentication comprising:
a client computer;
a plurality of authentication servers;
a network interconnecting the client computer and plurality of authentication servers;
software running on the client computer and plurality of authentication servers that cooperates to enter a password on the client, store a unique random value y_i on each of the servers, derive a group element (P) from the password, send a blinded password value (P^x) to the servers, retrieve blinded key shares (P^{xy_i}) from the servers, unblind and combine the shares to create a master key (K_m), and decrypt encrypted private data on the client computer using the master key (K_m).

2. The system recited in Claim 1 wherein the software operating on the client operates to validate the master key (K_m).

3. The system recited in Claim 1 wherein the software operating on the client operates to decrypt encrypted private data using the master key (K_m).

4. The system recited in Claim 2 wherein the software operating on the client operates to decrypt encrypted private data using the validated master key (K_m).

5. The system recited in Claim 2 wherein the software operating on the client operates to send proof of the validated master key (K_m) and each blinded password value (P^x) to the servers.

6. A method that provide for remote password authentication using a system comprising a client computer, a plurality of authentication servers, and a network interconnecting the client computer and plurality of authentication servers, the method comprising the steps of:
entering a password;
deriving group elements (P) from the password;
sending blinded password value (P^x) to the servers;

retrieving blinded key shares (P^{xy_i}) from the servers;

unblinding and combining the shares to create a master key (K_m); and

decrypting encrypted private data on the client computer using the master key (K_m).

7. The method recited in Claim 6 further comprising the step of validating the master key (K_m).

8. The method recited in Claim 6 wherein the software operating on the client operates to decrypt encrypted private data using the master key (K_m).

9. The method recited in Claim 7 further comprising the step of decrypting encrypted private data using the validated master key (K_m).

10. The method recited in Claim 7 further comprising the step of sending proof of the validated master key (K_m) and each blinded password value (P^x) to the servers.

11. A computer program embodied on a computer-readable medium for enabling remote password authentication in a multiple-server system comprising a client computer, a plurality of authentication servers, and a network interconnecting the client computer and plurality of authentication servers, the computer program comprising:

a code segment that enters a password;

a data storage area that contains a unique random value y_i on each of the servers,

a code segment that derives a group element (P) from the password;

a code segment that sends blinded password value (P^x) to the servers;

a code segment that retrieves blinded key shares (P^{xy_i}) from the servers;

a code segment that unblinds and combines the shares to create a master key (K_m); and

a code segment that decrypts encrypted private data on the client computer using the master key (K_m).

12. The computer program recited in Claim 11 further comprising a code segment that validates the master key (K_m).

13. The computer program recited in Claim 11 further comprising a code segment that decrypts encrypted private data using the master key (K_m).

14. The computer program recited in Claim 12 further comprising a code segment that decrypts encrypted private data using the validated master key (K_m).

15. The computer program recited in Claim 12 further comprising a code segment that sends proof of the validated master key (K_m) and the blinded password value (P^x) to the servers.

16. The system recited in Claim 1 wherein the software cooperates to:

maintain a count of bad login attempts, the number of recent amplifications, a list of recent P^x password amplification request values, and a list of timestamps associated with the list of recent password amplification request values on the server;

receives a blinded password (P^x) request

records the blinded password in a short-term list

checks a user account to see if it is locked;

creates a blinded key share ($P^{xy}{}_i$); and

sends the blinded key share to the client computer if it is unlocked.

17. The system recited in Claim 16 wherein the software:

records a timestamp value to note the time that the request was received;

periodically checks for stale requests which are determined when the difference between any timestamp value and the current time becomes greater than a specific period of time;

deletes corresponding password amplification request values and timestamps; and

increments the count of bad attempts.

18. The system recited in Claim 16 wherein, when a successful login occurs, the software:

sends a value of Q_A, equal to the password raised to a random power, along with any prior values for Q_A from earlier runs in the same login session, to each server in an encrypted message; and

authenticates this message using the master key K_m.

19. The method recited in Claim 6 further comprising the steps of

maintaining a count of bad login attempts, the number of recent amplifications, a list of recent P^X password amplification request values, and a list of timestamps associated with the list of recent password amplification request values on the server;

receiving a blinded password (P^X) request

recording the blinded password in a short-term list

checking a user account to see if it is locked;

creating a blinded key share (P^{XY_i}); and

sending the blinded key share to the client computer if it is unlocked.

20. The system recited in Claim 19 wherein the software:

records a timestamp value to note the time that the request was received;

periodically checks for stale requests which are determined when the difference between any timestamp value and the current time becomes greater than a specific period of time;

deletes corresponding password amplification request values and timestamps; and

increments the count of bad attempts.

21. The method recited in Claim 19 further comprising the steps of

sending the value of Q_A, equal to the password raised to a random power, along with any prior values for Q_A from earlier runs in the same login session, to each server in an encrypted message; and

authenticating this message using the master key K_m.

22. The computer program recited in Claim 11 further comprising a code segment that:

maintains a count of bad login attempts, the number of recent amplifications, a list of recent P^X password amplification request values, and a list of timestamps associated with the list of recent password amplification request values on the server;

receives a blinded password (P^X) request

records the blinded password in a short-term suspect list

checks a user account to see if it is locked;

creates a blinded key share (P^{XY_i}) if it is unlocked; and

sends the blinded key share to the client computer.

23. The computer program recited in Claim 22 further comprising a code segment that:

records a timestamp value to note the time that the request was received;

periodically checks for stale requests which are determined when the difference between any timestamp value and the current time becomes greater than a specific period of time;

deletes corresponding password amplification request values and timestamps; and

increments the count of bad attempts.

24. The computer program recited in Claim 22 further comprising a code segment that:

sends the value of Q_A, equal to the password raised to a random power, along with any prior values for Q_A from earlier runs in the same login session, to each server in an encrypted message; and

authenticates this message using the master key K_m.